

E8 Emulator

Additional Document for User's Manual

R0E000080KCE00EP8

Renesas Microcomputer Development Environment System
M16C Family / M32C/80 Series
Notes on Connecting the M32C/84, M32C/85, M32C/86, M32C/87
and M32C/88

User's Manual

Rev.4.00
November 01, 2006

Renesas Technology
www.renesas.com

Keep safety first in your circuit designs!

1. Renesas Technology Corp. puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage. Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of non-flammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corp. product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corp. or a third party.
2. Renesas Technology Corp. assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corp. without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor for the latest product information before purchasing a product listed herein.
The information described here may contain technical inaccuracies or typographical errors. Renesas Technology Corp. assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.
Please also pay attention to information published by Renesas Technology Corp. by various means, including the Renesas Technology Corp. Semiconductor home page (<http://www.renesas.com>).
4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corp. assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corp. semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corp. or an authorized Renesas Technology Corp. product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
6. The prior written approval of Renesas Technology Corp. is necessary to reprint or reproduce in whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
Any diversion or reexport contrary to the export control laws and regulations of Japan and/ or the country of destination is prohibited.
8. Please contact Renesas Technology Corp. for further details on these materials or the products contained therein.

Contents

Section 1	Specifications of the E8 Emulator	1
Section 2	Connecting the Emulator with the User System	3
Section 3	Pin Assignments of the E8 Connector	5
Section 4	Example of E8 Connection	7
Section 5	Notes on Using the E8 Emulator.....	15
Section 6	Setup the Debugger	23
Section 7	Applicable Tool Chain and Partner Tools	26

This user's manual is applicable to the E8 emulator software V.2.09 Release 00 or later.

Section 1 Specifications of the E8 Emulator

Table 1.1 shows the specifications of the M32C/80 Series E8 Emulator.

Table 1.1 Specifications of the M32C/80 Series E8 Emulator

Target MCU	M16C Family M32C/80 Series M32C/84, M32C/85, M32C/86, M32C/87 and M32C/88 Groups		
Usable operating mode	Single-chip mode, Memory expansion mode * Microprocessor mode is not supported.		
Break function	- Address match break, 8 points - PC break (up to 255 points) - Forcible break		
Trace function	Not available		
Flash memory programming function	Available		
User interface	Clock-synchronized serial (communicating via P64/P65/P66/P67)		
MCU resource to be used	- ROM size: 2 KB (alterable assigned address) - RAM size: 256 bytes (alterable assigned address) - Stack 20 bytes - UART 1 function and P64/P65/P66/P67 - Pins P50 and P55 - Address match interrupt		
Emulator power supply	Unnecessary (USB bus powered, power supplied from the PC)		
Interface with host machine	USB (USB 1.1, full speed) * Also connectable to host computers that support USB 2.0		
Power supply function	Can supply 3.3V or 5.0V to the target board (300mA, max)		
Power voltage	M32C/84	Normal-ver.	3.0V ± 0.3V or 5.0V ± 0.5V
	M32C/85	T-ver.	5.0V ± 0.5V
	M32C/86	-	3.0V ± 0.3V or 5.0V ± 0.5V
	M32C/87	-	3.0V ± 0.3V or 5.0V ± 0.5V
	M32C/88	-	5.0V ± 0.5V

Table 1.2 shows the operating environment of the E8 Emulator.

Table 1.2 Operating Environment

Temperature	Operating	: 10°C to 35°C
	Storage	: -10°C to 50°C
Humidity	Operating	: 35% RH to 80% RH, no condensation
	Storage	: 35% RH to 80% RH, no condensation
Vibration	Operating	: 2.45 m/s ² max.
	Storage	: 4.9 m/s ² max.
	Transportation	: 14.7 m/s ² max.
Ambient gas	No corrosive gas	

Section 2 Connecting the Emulator with the User System

Before connecting an E8 emulator with the user system, a connector must be installed in the user system so that a user system interface cable can be connected. When designing the user system, refer to Figure 3.1, Pin Assignments of the E8 Connector, and Figure 4.1, Example of E8 Connection, shown in this manual.

Before designing the user system, be sure to read the E8 emulator user's manual and the hardware manual for related MCUs.

Table 2.1 shows the recommended connector for the emulator.

Table 2.1 Recommended Connector

	Type Number	Manufacturer	Specifications
14-pin connector	7614-6002OO*	3M Limited	14-pin straight type

*OO indicates coat specification

Connect pins 2, 6, 10, 12, and 14 of the user system connector to GND firmly on the PCB. These pins are used as electrical GND and to monitor the connection of the user system connector. Note the pin assignments of the user system connector.

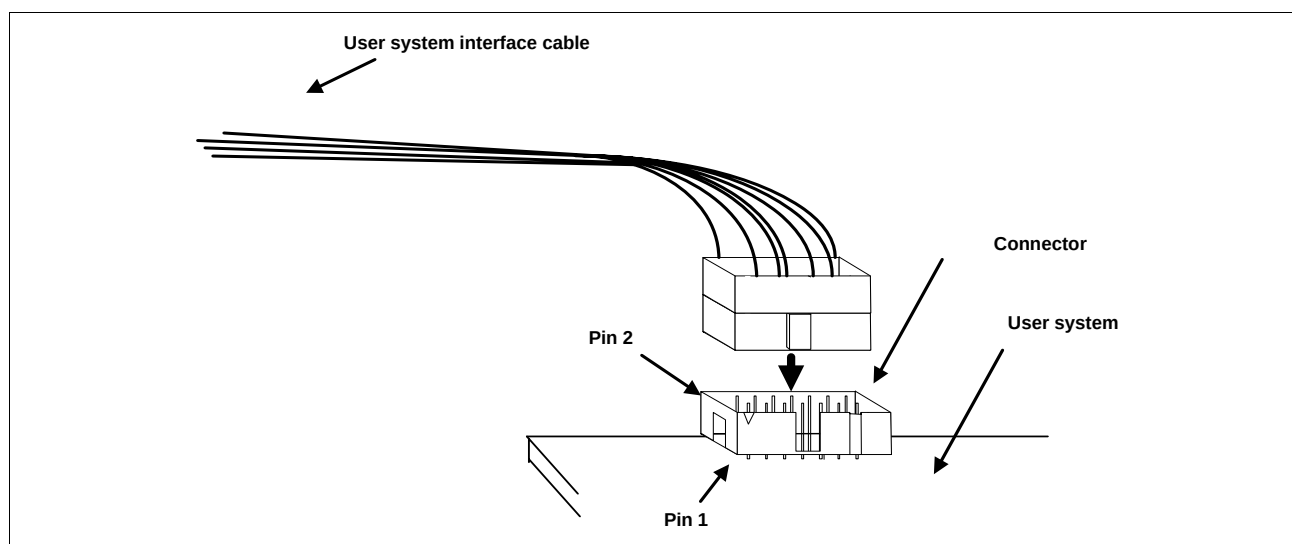


Figure 2.1 Connecting the User System Interface Cable to the User System

- Notes:
1. Do not place any components within 3 mm of the connector.
 2. When using the E8 emulator as a programmer, connect the E8 emulator to the user system in the same way.

Section 3 Pin Assignments of the E8 Connector

Figure 3.1 shows the pin assignments of the connector.

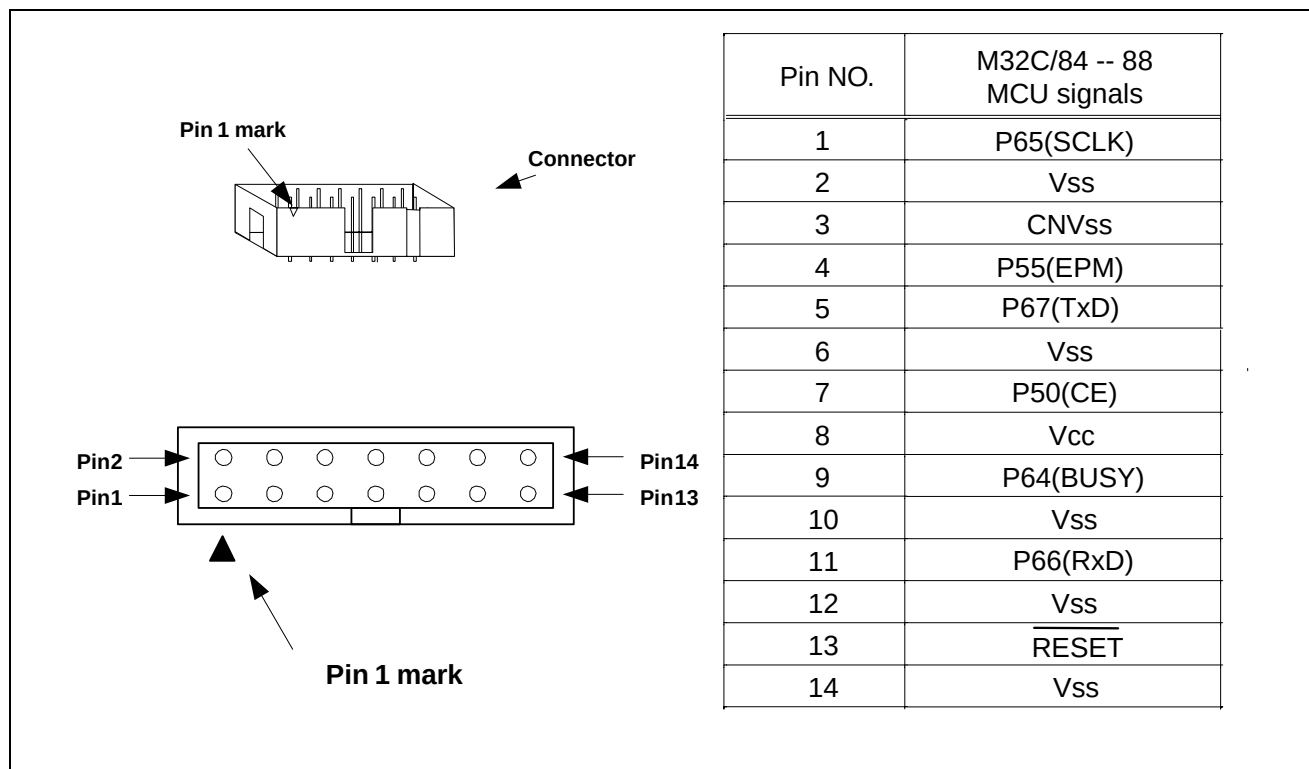


Figure 3.1 Pin Assignments of the E8 Connector

Note:

Pin 14 is used to check for a connection between the E8 and the user system, and is not directly connected to Vss inside the E8. Make sure that pin 14 and other pins 2, 6, 10, and 12 are connected to Vss.

Section 4 Example of E8 Connection

The connecting examples are shown as follows.

When using the emulator as a programmer, the specification of connection between the E8 and the MCUs is the same as shown in Figure 4.1.

(1) In single-power supply and single-chip mode

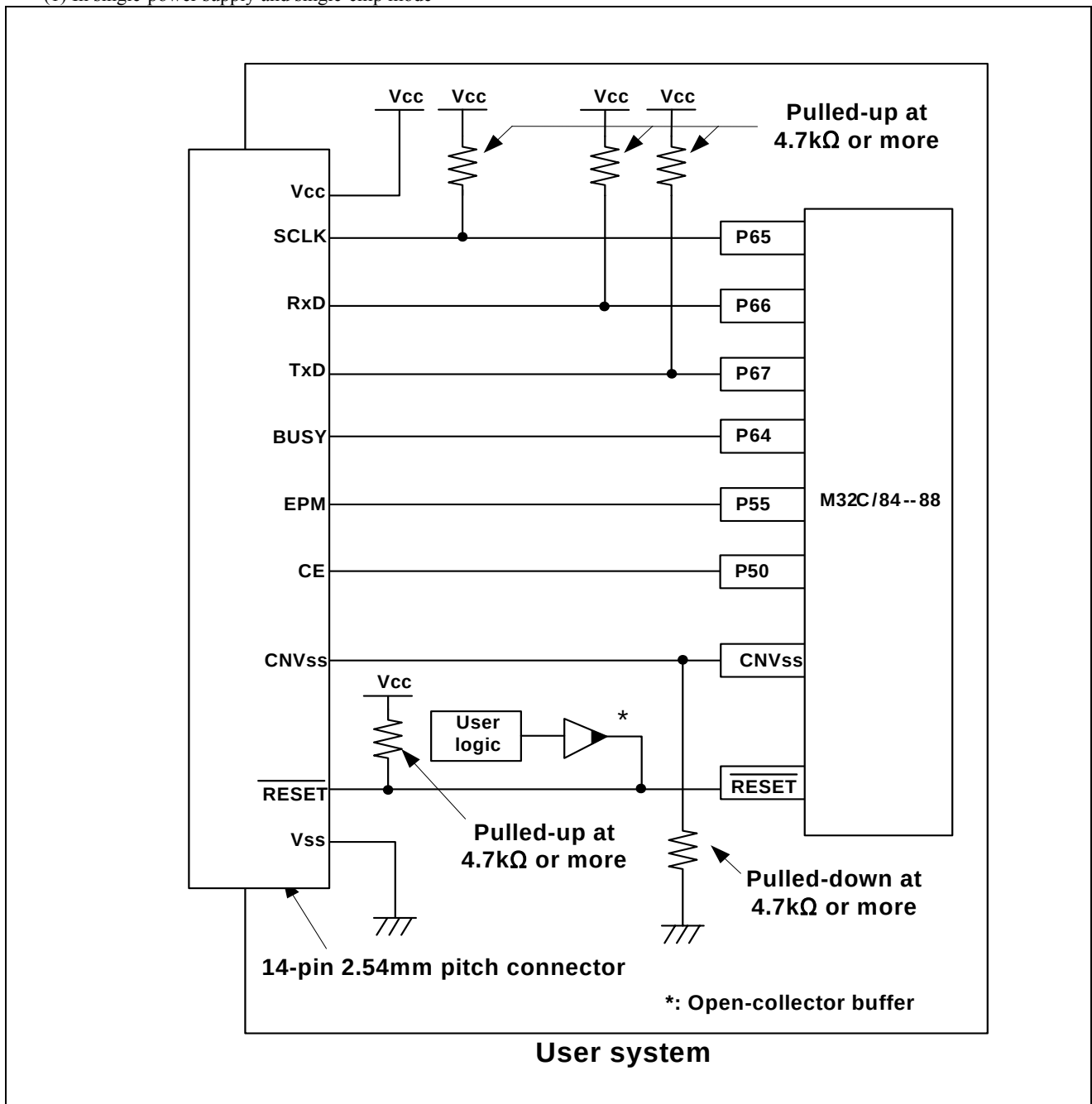


Figure 4.1 Example of E8 Connection (Single-power Supply and Single-chip Mode)

(2) In single-power supply and memory expansion mode

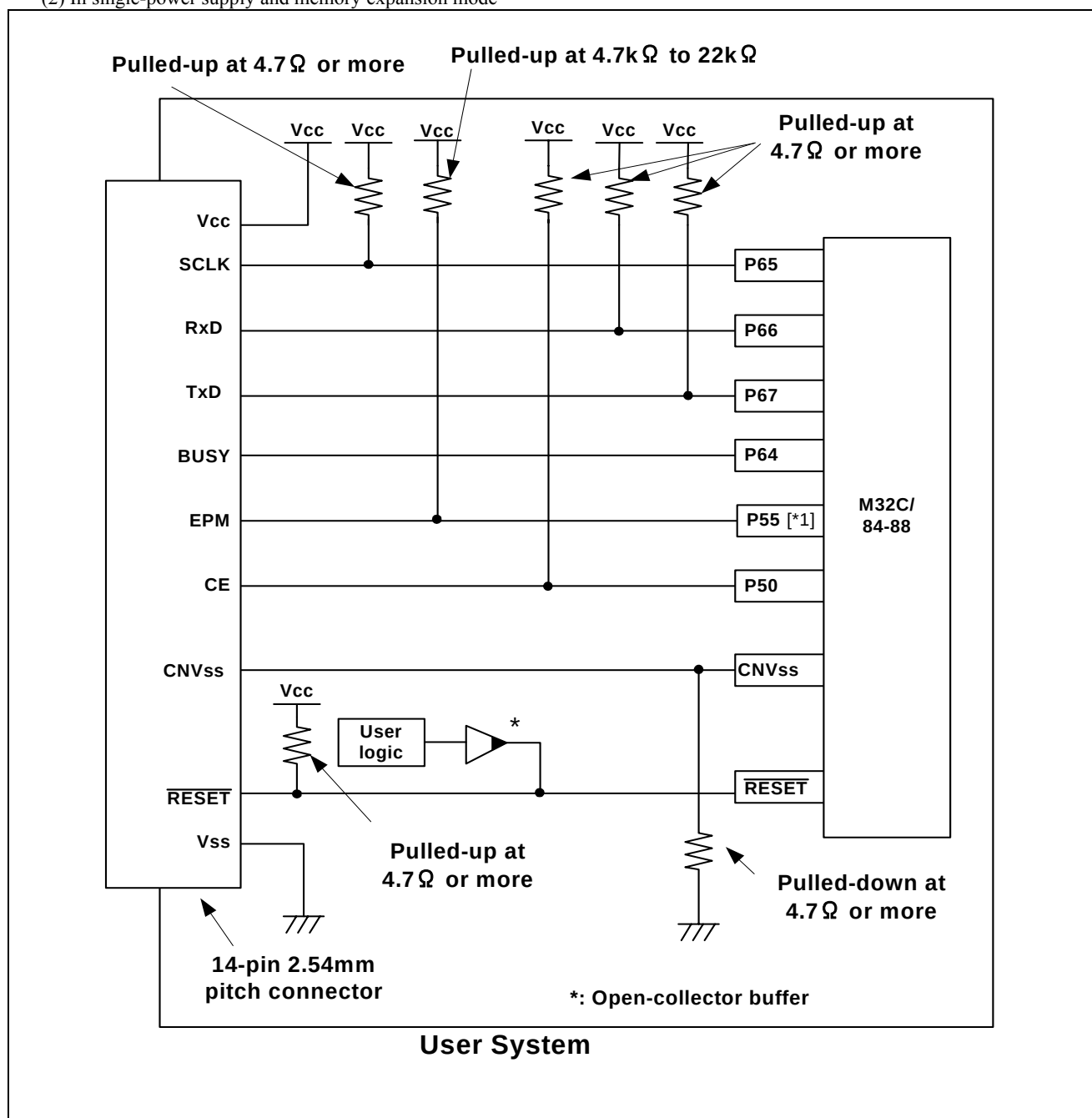


Figure 4.2 Example of E8 Connection (Single-power Supply and Memory Expansion Mode)

[*1]: The $\overline{\text{HOLD}}$ signal cannot be used. Pulled-up P55 on the user system.

(3) In dual-power supply and single-chip mode

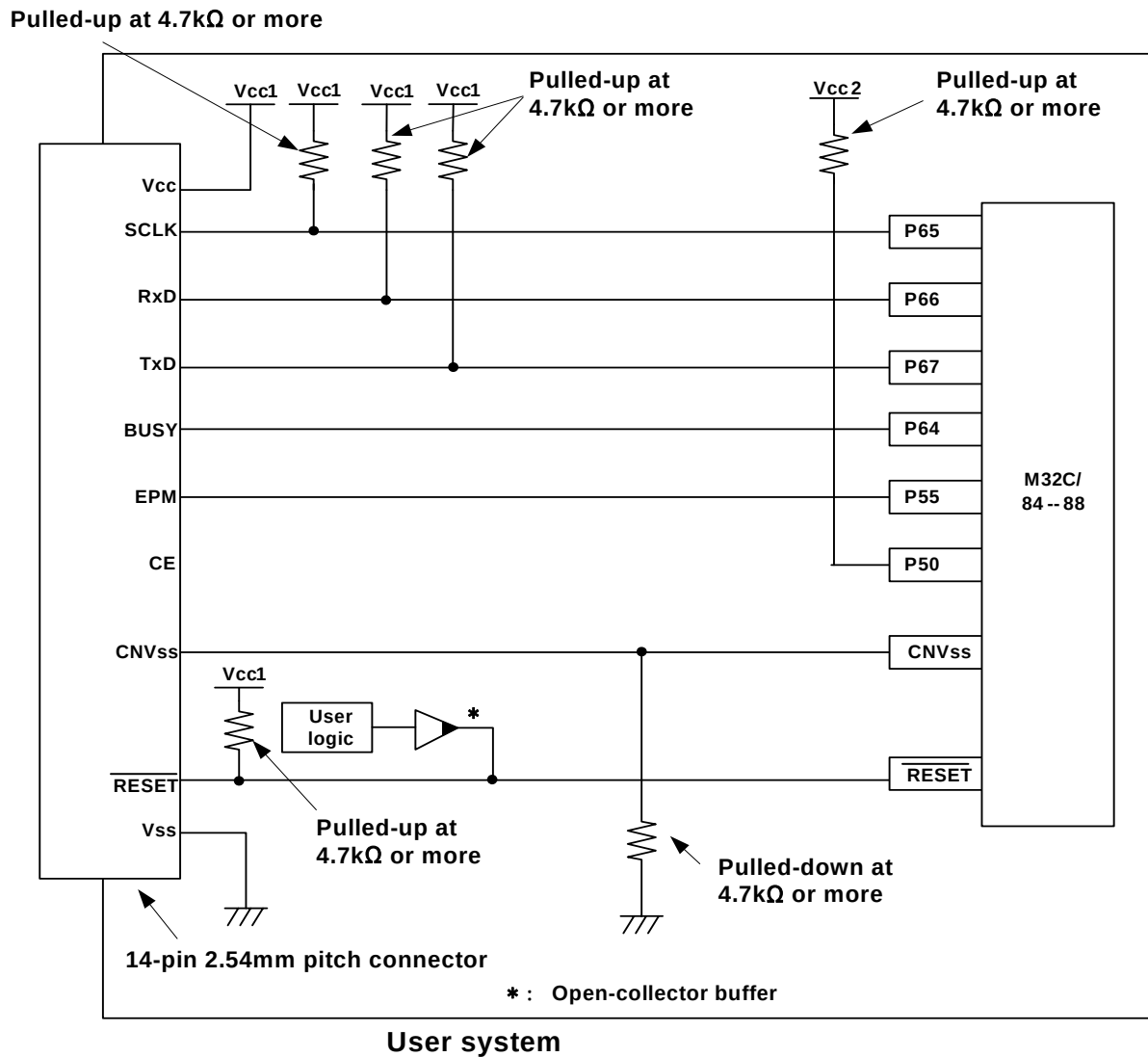


Figure 4.3 Example of E8 Connection (Dual-power Supply and Single-chip Mode)

(4) In dual-power supply and memory expansion mode

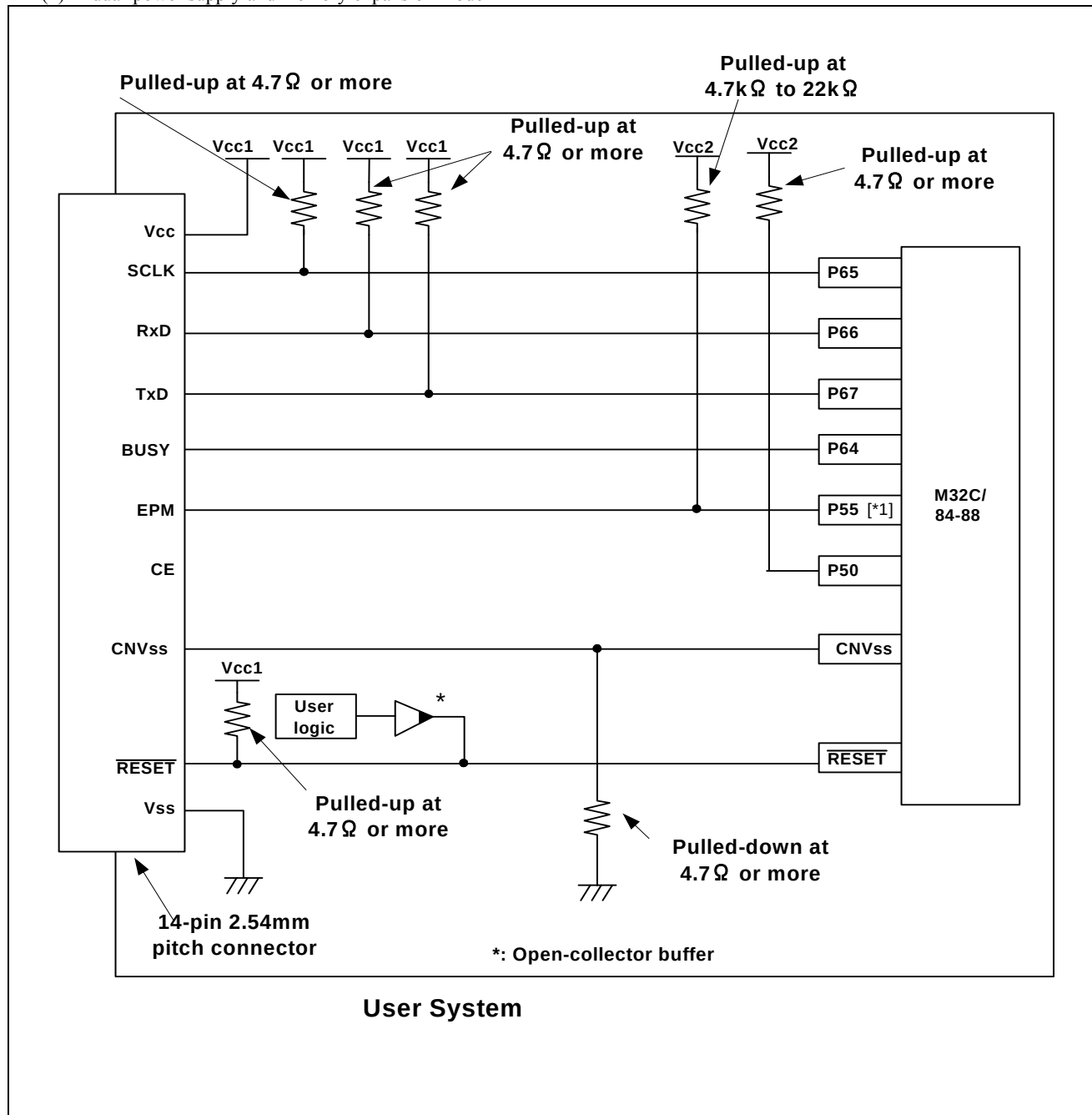


Figure 4.4 Example of E8 Connection (Dual-power Supply and Memory Expansion Mode)

[*1]: The $\overline{\text{HOLD}}$ signal cannot be used. Pulled-up P55 on the user system.

Notes: 1. P64, P65, P66 and P67 pins are used by the E8 emulator. Connect the E8 emulator to the MCU pins. For MCU pins P65, P66 and P67, pull up and connect to the E8 emulator.

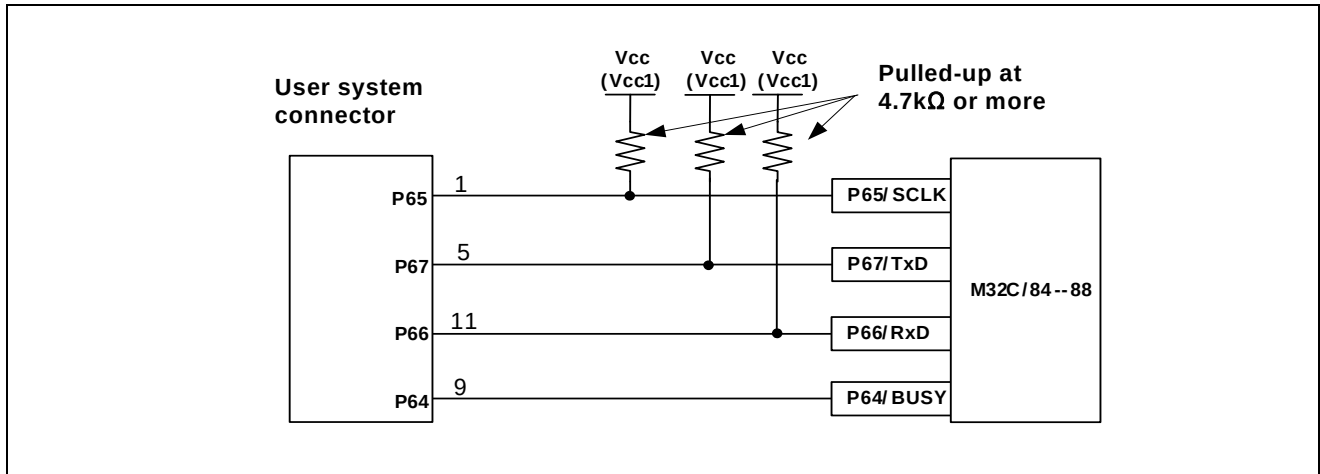


Figure 4.5 Connection of E8 Emulator and MCU

2. The E8 emulator uses the P50 and P55 pins for the MCU control. Connect the E8 emulator to the MCU pins.

(1) In single-power supply and single-chip mode

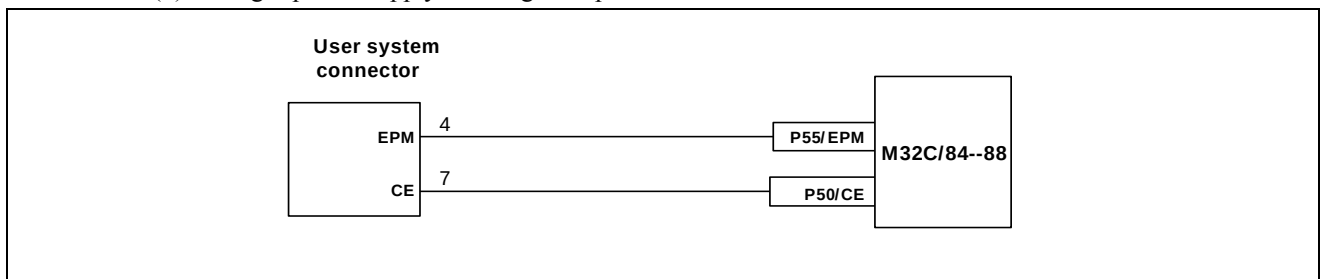


Figure 4.6 Connection of E8 Emulator and P50 and P55 Pins (Single-power Supply and Single-chip Mode)

(2) In single-power supply and memory expansion mode

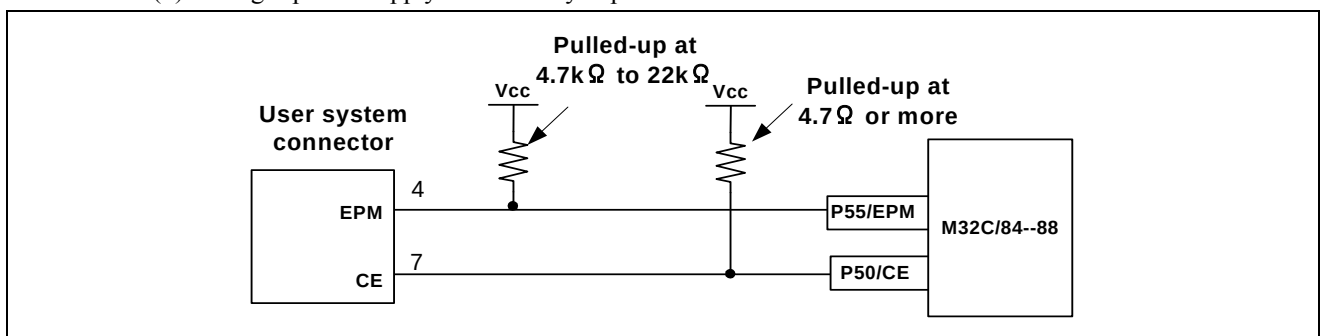


Figure 4.7 Connection of E8 Emulator and P50 and P55 Pins (Single-power Supply and Memory Expansion Mode)

Note: The HOLD signal cannot be used. Pulled-up P55 on the user system..

(3) In dual-power supply and single-chip mode

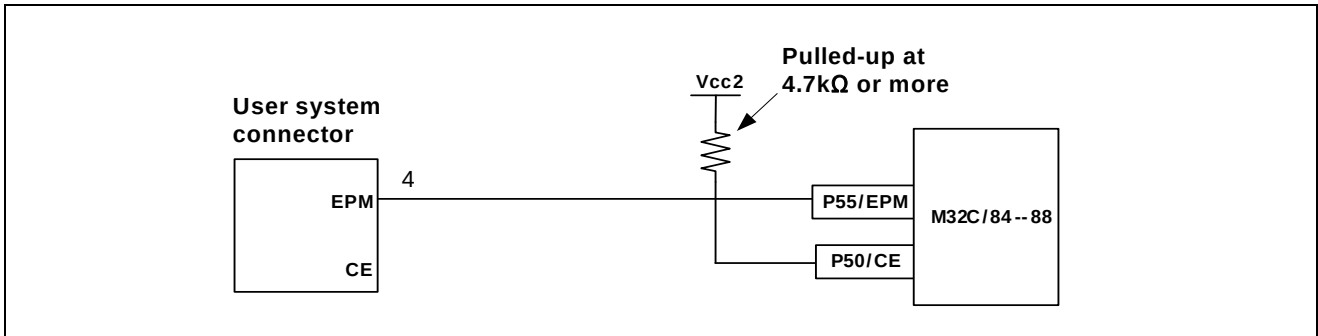


Figure 4.8 Connection of E8 Emulator and P50 and P55 Pins
(Dual-power Supply and Single-chip Mode)

(4) In dual-power supply and memory expansion mode

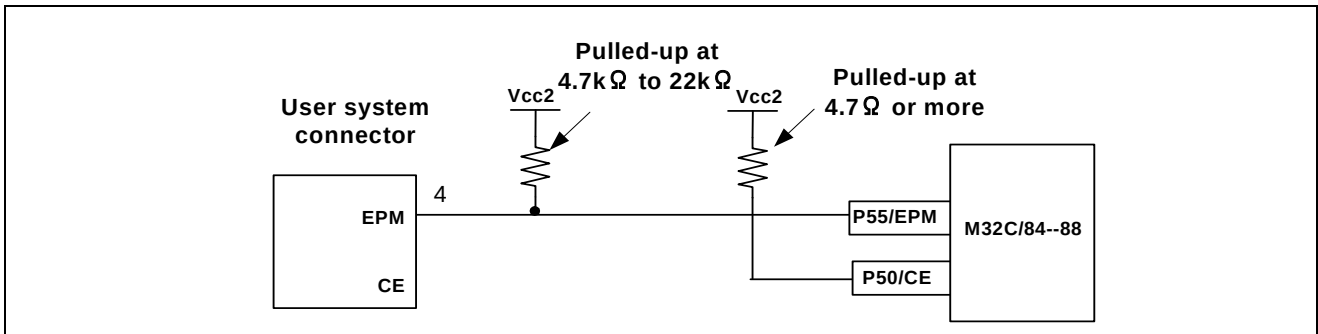


Figure 4.9 Connection of E8 Emulator and P50 and P55 Pins
(Dual-power Supply and Memory Expansion Mode)

Note: The $\overline{\text{HOLD}}$ signal cannot be used. Pulled-up P55 on the user system.

3. The E8 emulator uses the CNVss pin for the MCU control. Pulled-down the E8 emulator and the MCU pins to connect the E8 emulator

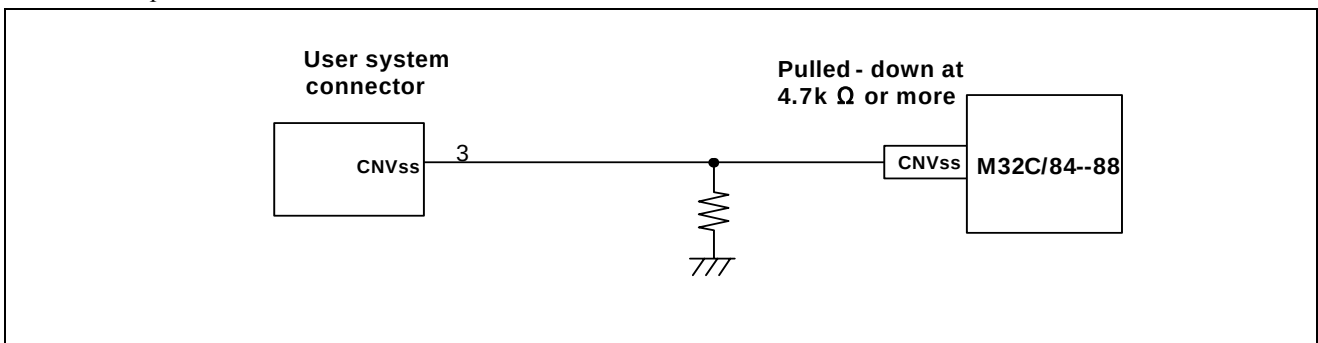


Figure 4.10 Connection of E8 Emulator and CNVss Pin

4. The $\overline{\text{RESET}}$ pin is used by the E8 emulator. Therefore use an open-collector output buffer or a CR reset circuit as the reset circuit of the user system. The recommended pull-up value is 4.7k Ω or more. The MCU can be reset by outputting “L” from the E8 emulator. However, if a reset circuit on the user system is the H-output type reset IC, it cannot be set “L” in the reset circuit on the user system and the E8 emulator will not operate normally.

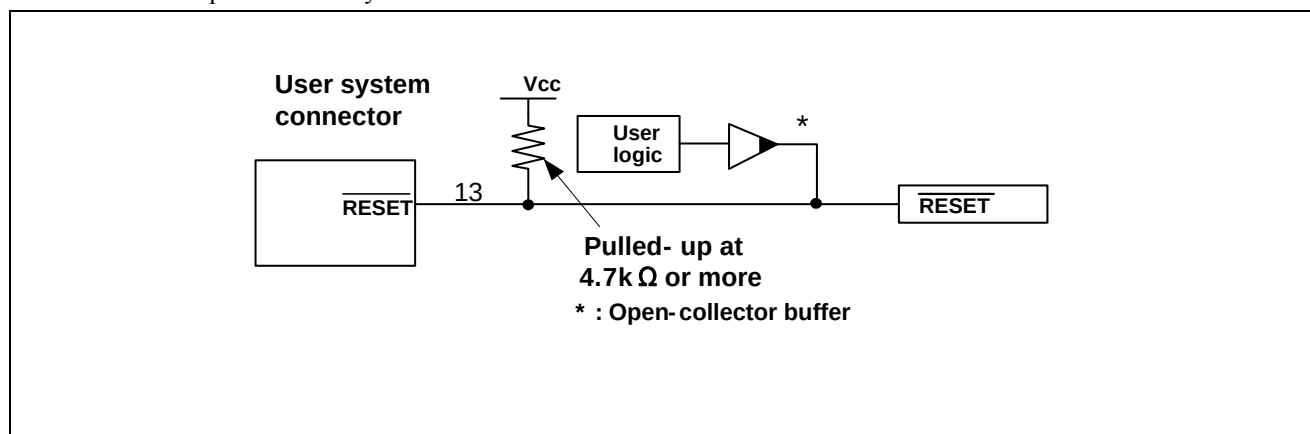


Figure 4.11 Example of a Reset Circuit

5. Connect Vss and Vcc with the Vss and Vcc (Vcc1) of the MCU, respectively.
6. Connect nothing with N.C.
7. The amount of voltage permitted to input to Vcc (Vcc1, Vcc2) must be within the guaranteed range of the microcomputer.
8. If the $\overline{\text{NMI}}$ pin interrupt are unused, make sure the $\overline{\text{NMI}}$ pin is pulled-up to the Vcc (Vcc1) pin through a resistor.
9. Pin 14 is used to check for a connection between the E8 and the user system, and is not directly connected to Vss inside the E8. Make sure that pin 14 and other pins 2, 6, 10, and 12 are connected to Vss.

10. Figure 4.12 shows the interface circuit in the E8 emulator. Use this figure as a reference when determining the pull-up resistance value.

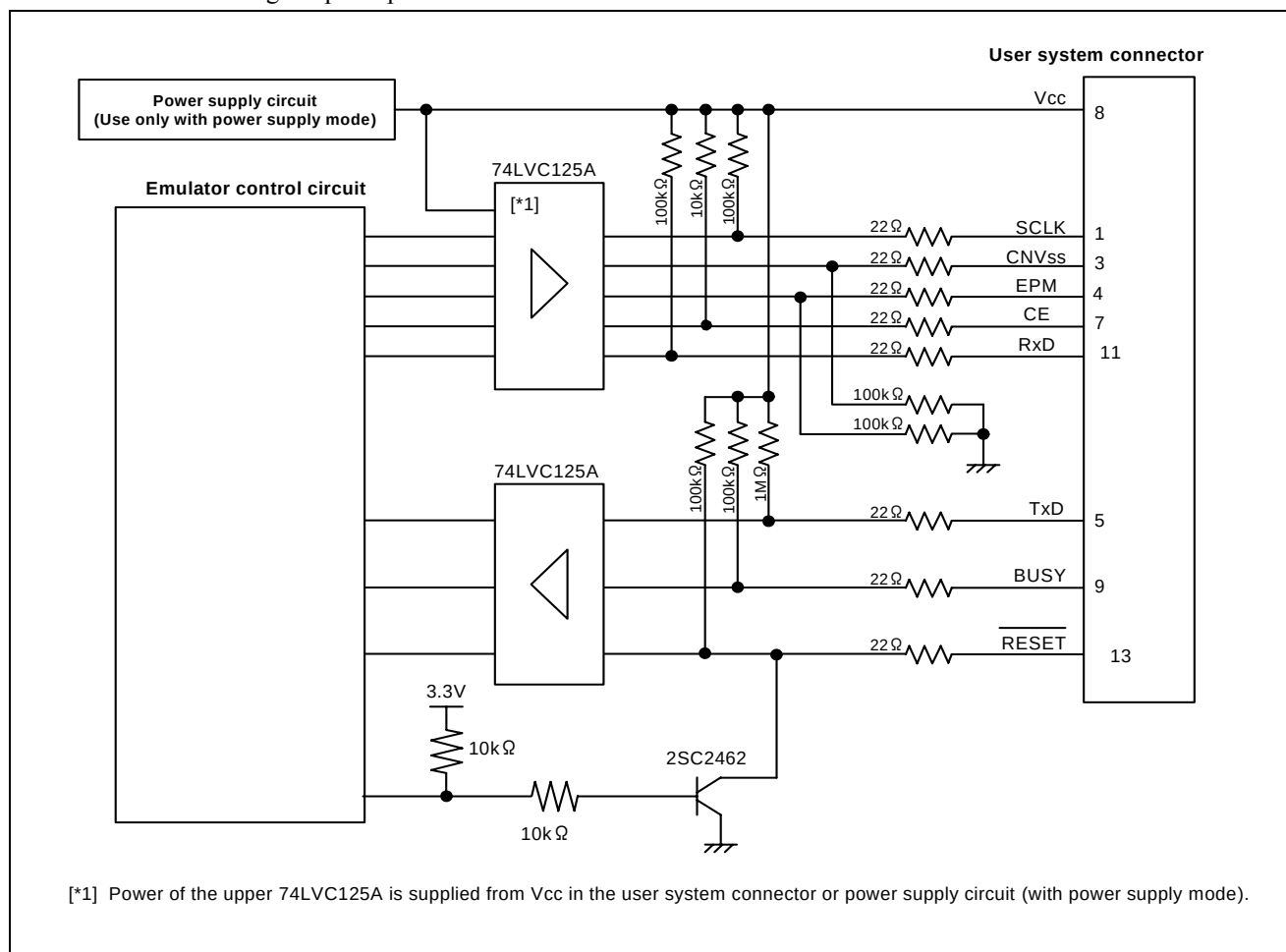


Figure 4.12 Interface Circuit in the Emulator (Reference)

Section 5 Notes on Using the E8 Emulator

1. Program area for the E8 emulator

Table 5.1 lists the program area for the E8 emulator.

Do not change this area. If this area is changed, the E8 emulator will not operate normally. In this case, disconnect to the debugger and then reconnect.

Table 5.1 Program Area for the E8 Emulator

Group	Type Number	ROM Size		RAM Size	Program Area for E8 Emulator		
		Programming Area	Data Area		Vector Area	ROM Area	RAM Area
M32C/84	M30843FW	320KB	4 KB	24KB	FFFFE8h--FFFFEBh, FFFFFCh--FFFFFh,	2 KB of the programming area [*1]	256 bytes [*1]
	M30843FH	384KB					
	M30843FJ	512KB					
	M30845FW	320KB					
	M30845FH	384KB					
	M30845FJ	512KB					
M32C/85	M30853FW	320KB		24KB			
	M30853FH	384KB					
	M30853FJ	512KB					
	M30855FW	320KB					
	M30855FH	384KB					
	M30855FJ	512KB					
M32C/86	M30865FJ	512KB		24KB			
M32C/87	M30879FL	1MB		48KB			
	M3087BFL			24KB			
	M30873FH	384KB					
	M30875FH						
	M30876FJ	512KB		48KB			
	M30878FJ						
	M30879FK	768KB					
	M3087BFK						
M32C/88	M30880FW	320KB		18KB			
	M30882FW						
	M30880FH	384KB					
	M30882FH						
	M30880FJ	512KB					
	M30882FJ						

*1: When starting up the debugger, the [Emulator Setting] dialog box shown in Figure 5.1 is displayed. Specify an area which is not used in the user system.

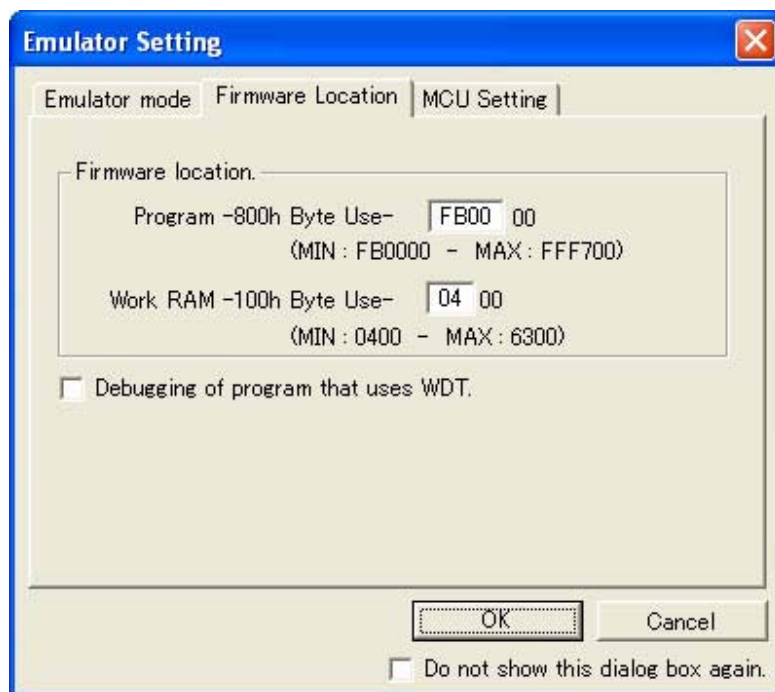


Figure 5.1 [Firmware Location] Tab of [Emulator Setting] Dialog Box

- When the emulator system is activated, and CPU is reset from the debugger, it initializes the general registers and part of the flag registers as shown in Table 5.2.

Table 5.2 Register Initial Values at Emulator Power-On

Register	Initial Value
R0 to R3 (bank 0, 1)	0000h
A0, A1 (bank 0, 1)	000000h
FB (bank 0, 1)	000000h
SB (bank 0, 1)	000000h
PC	Reset vector value in the vector address table
INTB (bank 0, 1)	000000h
USP	000000h
ISP	Work RAM Address for the E8 emulator + 100h *
SVF	000000h
SVP	000000h
VCT	000000h
DMD0, DMD1	00h
DCT0, DCT1	0000h
DRC0, DRC1	0000h
DMA0, DMA1	000000h
DRA0, DRA1	000000h
DSA0, DSA1	000000h

Note: The Work RAM address for the E8 emulator is specified in [Firmware Location] of [Emulator Setting] dialog box.

3. The E8 emulator controls the MCUs by using the P50, P55, P64, P65, P66, P67, RESET and CNVss pins.
4. The emulator uses up to 20-byte stack pointer when a user program breaks. Accordingly, reserve the 20-byte addresses for the stack area.
5. SFR used by the program for the E8 emulator
As the SFR listed in Table 5.3 is used by the program for the E8 emulator, do not change a value. Otherwise, the E8 emulator cannot be controlled. Note that UART1 transmit interrupt control register S1TIC and UART1 receive interrupt control register S1RIC always read out the value of using the emulator. And they are not initialized by selecting [Debug] -> [Reset CPU] or with the RESET command. If their contents are referred to, a value that has been set in the program for the E8 emulator will be read.

Table 5.3 SFR Used by Program for E8 Emulator

Address	Register	Symbol	Bit	Notes on using the E8 emulator
0009h	Address match interrupt enable register	AIER	All bits	[*1]
0010h - 0012h	Address match interrupt register 0	RMAD0	All bits	[*1]
0014h - 0016h	Address match interrupt register 1	RMAD1	All bits	[*1]
0018h - 001Ah	Address match interrupt register 2	RMAD2	All bits	[*1]
001Ch - 001Eh	Address match interrupt register 3	RMAD3	All bits	[*1]
0028h - 002Ah	Address match interrupt register 4	RMAD4	All bits	[*1]
002Ch - 002Eh	Address match interrupt register 5	RMAD5	All bits	[*1]
0038h - 003Ah	Address match interrupt register 6	RMAD6	All bits	[*1]
003Ch - 003Eh	Address match interrupt register 7	RMAD7	All bits	[*1]
02E8h	UART1 transmit/receive mode register	U1MR	All bits	[*1]
02EAh, 02EBh	UART1 transmit buffer register	U1TB	All bits	[*1]
02ECh	UART1 transmit/receive control register 0	U1C0	All bits	[*1]
02EDh	UART1 transmit/receive control register 1	U1C1	All bits	[*1]
02EEh, 02EFh	UART1 receive buffer register	U1RB	All bits	[*1]
03C0h	Port P6 register	P6	Bits 4, 5, 6 and 7	[*2]
03C2h	Port P6 direction register	PD6	Bits 4, 5, 6 and 7	[*2]
03B0h	Function select register A0	PS0	Bits 4, 5, 6 and 7	[*2]

- *1 Do not change the value of the register.
 *2 Do not change the value of the bits listed above. When operating this register, change it by a bit operating instruction, etc. in order to avoid changing the value of relevant bits.

6. Interrupts used by the E8 emulator program
The BRK2 instruction interrupt, address match interrupt and single-step interrupt are used by the E8 emulator program. Therefore, make sure the user program does not use these interrupts. The E8 emulator changes these interrupt vector values to the values to be used by the emulator. It is not a problem if the interrupt vector values are written in the user program.

7. Debugging of the watchdog timer

When debugging the user program using the watchdog timer, select the [Debugging of program that uses WDT.] check box in the [Firmware Location] of [Emulator Setting] dialog box.

If this box is unchecked and the watchdog timer is enabled during debugging, the E8 emulator will be out of control. When selecting [Debugging of program that uses WDT.] check box, the program for the E8 emulator refreshes the watchdog timer during the E8 emulator program execution. If a memory is accessed by the memory reference or modification, the watchdog timer will be refreshed by the program for the E8 emulator. Note that the operation timing is different from the actual one.

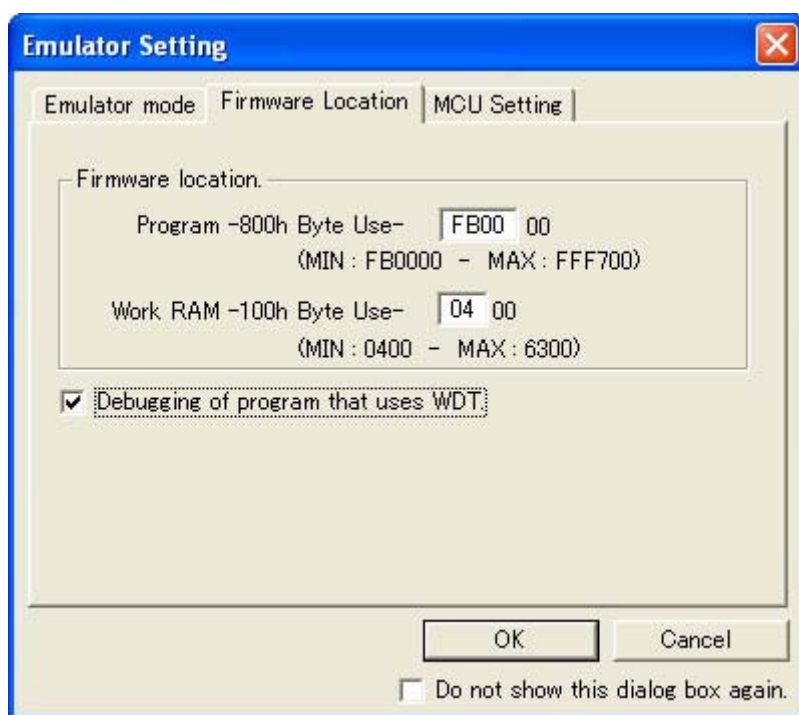


Figure 5.2 [Firmware Location] Tab of [Emulator Setting] Dialog Box

8. ID code of flash memory

This is the function of the MCUs which prevents the flash memory from reading out by other than the user. The 7 bytes ID code in Table 5.4 written to the flash memory of the MCUs have to match with the ID code displayed in Figure 5.3 [ID Code verification] dialog box at the debugger startup, otherwise the debugger cannot be started up. Note that when the ID code is FFh, FFh, FFh, FFh, FFh, FFh, FFh, ID code is considered not set up. In this case, the ID code is automatically authenticated and [ID Code verification] dialog box is not displayed.

When debugging in [Erase Flash and Connect] mode or [Keep Flash and Connect] mode, the ID code, FFh, FFh, FFh, FFh, FFh, FFh, FFh is written into the ID code area regardless of the contents of the user program. In [Program Flash] mode, the value to be written in the ID code area depends on the contents of the user program.

Table 5.4 ID Code Storage Area of M32C

Address	Description
FFFFDFh	First byte of ID code
FFFFE3h	Second byte of ID code
FFFFEBh	Third byte of ID code
FFFFEFh	Fourth byte of ID code
FFFFF3h	Fifth byte of ID code
FFFFF7h	Sixth byte of ID code
FFFFFBh	Seventh byte of ID code

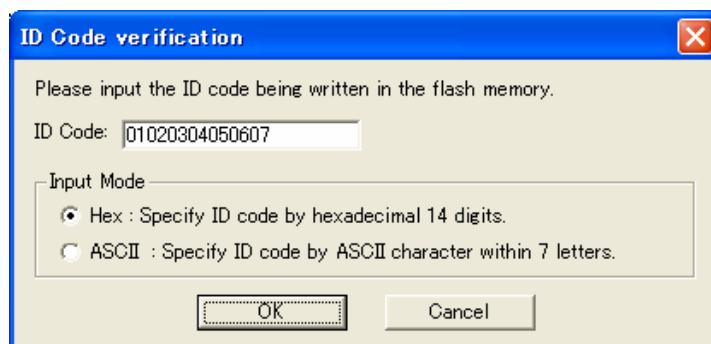


Figure 5.3 [ID Code verification] Dialog Box

[Note on Program Flash mode]

When the ID code is specified by the -ID option of the lmc308, download the MOT file or HEX file. When the X30 file is downloaded, the ID code is not effective. When downloading the X30 file, specify the ID code using an assembler directive command such as “.BYTE”. The file to which the ID code specified by the assembler directive command “.ID” is output varies depending on the version of the assembler. For details, refer to the user’s manual of the assembler.

9. Operation clock while the user program remains idle

While the user program remains idle, the E8 emulator program changes the main clock divide-by-N value as it runs.

10. Reset

The reset vector is used by the E8 emulator program. If the MCU is reset while executing the user program, control is transferred to the E8 emulator program and the user program is made to stop.

Do not use the hardware reset 2, software reset and watchdog timer reset, otherwise the E8 emulator will not operate normally.

11. Memory access during emulation execution

When referring or modifying the memory contents, the user program is temporarily halted. For this reason, real-time emulation cannot be performed. When the real-time emulation is necessary during the program operation, firstly disable the automatic update in the watch window or fix the display in the memory window so that the memory access will not occur during execution.

12. When the E8 does not supply power to the user system, the E8 emulator consumes the power voltage of the user system from several mA to over 10 mA. This is because the user power supply drives one 74LVC125A to make the communication signal level match the user system power supply voltage.

13. When debugging, the flash memory is frequently re-written by the E8 emulator. Therefore, do not use an MCU that has been used for debugging.

Also, as the program for the E8 emulator is written into the MCU while debugging, save the contents of the MCU's flash memory that have been used for debugging. Do not use them as the ROM data for products.

14. NMI interrupt

If NMI interrupts are to be used, be sure to take the necessary measures before executing the user program by, for example, disabling automatic updates of the watch window and freezing the display of the memory window in order to ensure that no memory accesses by the debugger will occur during user program execution.

If an NMI interrupt occurs while the user program remains idle or when memory contents are referenced or changed during user program execution, device operation becomes uncontrollable by the E8 emulator.

15. DMACII transfer complete interrupt

If DMACII transfer complete interrupts are to be used, be sure to take the necessary measures before executing the user program by, for example, disabling automatic updates of the watch window and freezing the display of the memory window in order to ensure that no memory accesses by the debugger will occur during user program execution.

If an DMACII transfer complete interrupt occurs while the user program remains idle or when memory contents are referenced or changed during user program execution, device operation becomes uncontrollable by the E8 emulator.

16. Reserved area

The addresses not specified in the Hardware Manual for M32C/84, M32C/85, M32C/86, M32C/87 and M32C/88 Groups are reserved area. Do not change the contents. Otherwise, the E8 emulator cannot be controlled.

17. Debugging in the stop mode or wait mode

When debugging in the stop or wait mode, the program cannot be stopped by the E8 emulator. If you attempt to stop the program during the stop or wait mode, the emulator will be uncontrollable.

Do not operate the window until the program stops at the breakpoint by setting the breakpoint at the processing unit where the stop mode or wait mode is cancelled. When using the stop mode or wait mode on a user program, firstly disable the automatic update in the watch window or fix the display in the memory window so that the memory access will not occur during execution.

18. Peripheral I/Os during a break

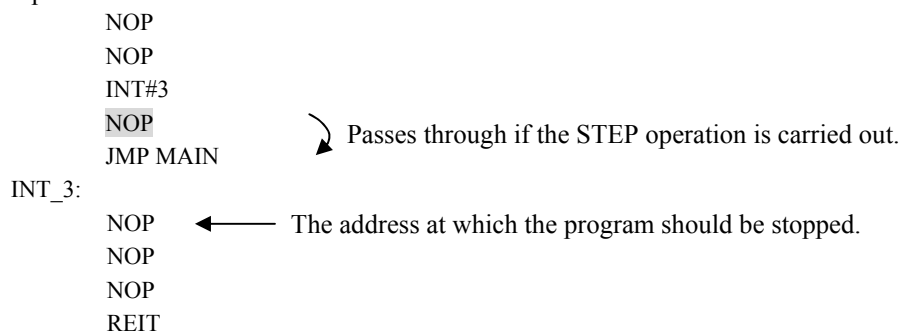
During a break, although interrupts are not accepted, peripheral I/Os continue to be operated. For example, a timer interrupt is not accepted although counting a timer is continued when a user program is stopped by a break after operating a timer.

19. Exceptional step operation

a) Software-interrupt instruction

STEP operation cannot be performed by continuously executing the internal processing of instructions (undefined, overflow, BRK, and INT) which generates a software interrupt.

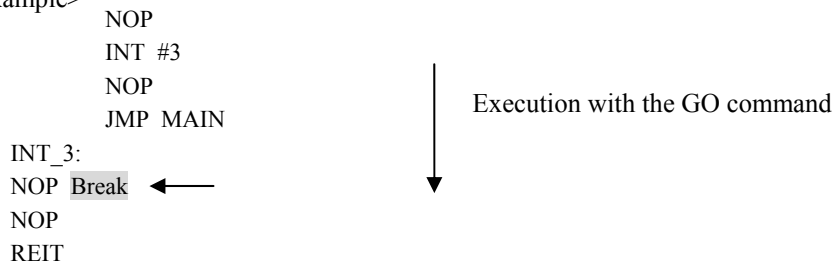
<Example> INT instruction



b) INT instruction

Debugging of the program using the INT instruction should be used with the GO command by setting a PC break for the internal processing of the INT instruction.

<Example>



20. "Go to cursor" function

The "Go to cursor" function is realized by using an address match break. Therefore, when you execute the "Go to cursor" command, all the address match breaks you set become invalid, while all the PC breaks remain valid.

21. Note on PC break point

When downloading a user program after changing it, the address setting of a PC break may not be corrected normally depending on the changes. After downloading a user program, please check the setting of a PC break by event point window and reset it.

22. Note on debugging in CPU rewrite mode

When debugging in CPU rewrite mode, do not rewrite the CPU's block 0 area (addresses FFF000h – FFFFFFFh) and block containing the program for the E8 emulator. If these areas are rewritten, the E8 emulator will run out of control.

Do not halt the user program after setting the CPU rewrite mode until releasing it. If you do so, the E8 emulator may run out of control. Cancel the automatic renewal in the watch window in advance and select fixing display in the memory window to prevent a memory access from occurring while executing the user program.

To check the data after executing the CPU rewrite mode, halt the program after releasing the CPU rewrite mode and see the memory window etc. When rewriting the flash memory in the program area, select Menu of the High-performance Embedded Workshop – [Setup] -> [Emulator] -> [System...] to open [Configuration] dialog box. In this dialog box, change [Flash memory synchronization] setting to [Flash memory to PC] and set off the cash of the debugger. In this setting, the flash memory is read whenever a break occurs, which takes some time. Use it with [Disable] setting except when debugging in CPU rewrite mode.

23. Notes on rewriting the flash memory

Do not reset the MCUs when rewriting the flash memory. The flash memory is completed to rewrite when the “Flash memory write end” is displayed in the output window of the High-performance Embedded Workshop. If the MCU is reset during rewriting the flash memory, the user program or the program for the E8 emulator may be disrupted

The followings indicate when the flash memory rewrite occurs.

- When downloading the user program
- After the user program starts with setting up PC break on the flash memory
- After the user program starts with canceling PC break on the flash memory
- After the user program starts with rewriting the value of flash memory in the memory window

24. Notes on the E8 emulator power supply

When writing reliability required program with the E8 emulator under the mass production, do not use the E8 emulator power supply function. Separately supply appropriate voltage according to the MCU programming to the user system. Since the supplied voltage from the E8 emulator depends on the quality of the USB power supply of the PC, its precision is not guaranteed.

Note that when debugging the system which operates the MCU with dual-power supply, the power cannot be supplied from the E8.

25. Notes on the emulator setup switch

Use the emulator setup switch with the factory setting (upper side “1”).

Section 6 Setup the Debugger

1. Emulator Setting dialog box

[Emulator Setting] dialog box is provided for setting the items that need to be set when the debugger starts up. The contents set from this dialog box other than “Power supply” are also effective the next time the debugger starts.

When starting up the debugger first time after creating the new project work space, [Emulator Setting] dialog box is displayed with the Wizard.

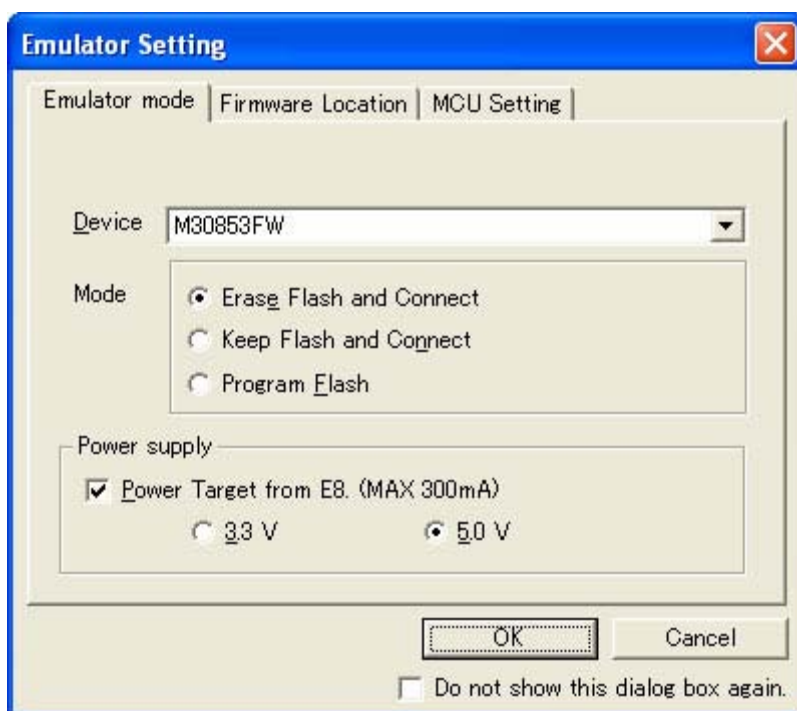


Figure 6.1 [Emulator Setting] Dialog Box.

To keep the [Emulator Setting] dialog box closed the debugger is started next time, check "Do not show this dialog box again." at the bottom of the [Emulator Setting] dialog box. You can open the [Emulator Setting] dialog box using either one of the following methods:

- After the debugger gets started, select Menu - [Setup] -> [Emulator] -> [Emulator Setting (E)...].
- Start Debugger while holding down the Ctrl key.

When you check the "Do not show this dialog box again.", the E8 does not supply power to the user system.

2. Emulator mode Tab

The selection of the device, the specification of the mode, and the setting of the power supply are done in the [Emulator mode] tab of the Emulator Setting dialog box.

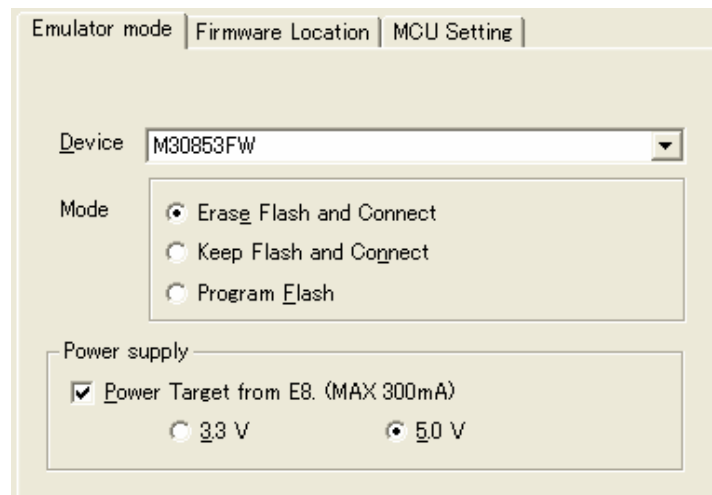


Figure 6.2 [Emulator mode] Tab

[Device]

Select the name of the MCUs to be used from the [Device] drop-down list box.

[Mode]

- Erase Flash and Connect

When starting up the debugger, erase the flash memory data of the MCUs. The program for the E8 emulator is written at the same time.

- Keep Flash and Connect

When starting up the debugger, keep the flash memory data of the MCUs. Note that the area for the E8 emulator program and the E8 emulator used vector area will be changed.

- Program Flash

Select this mode when using the E8 emulator as a simple programmer. On downloading write only the user program (The program for the E8 emulator is not written). Therefore, the program cannot be debugged in this mode.

[Power supply]

When [Power Target from E8. (MAX 300mA)] is checked, power will be supplied to the user system up to 300mA. Note that when debugging the system which operates the MCU with dual-power supply, the power cannot be supplied from the E8.

3. Firmware Location Tab

For details, see “1. Program area for the E8 emulator” and “7. Debugging of the watchdog timer” of the “Section 5. Notes on Using the E8 Emulator”.

4 MCU Setting Tab

In the [MCU Setting] tab, set the operating condition of MCU used in the user system.

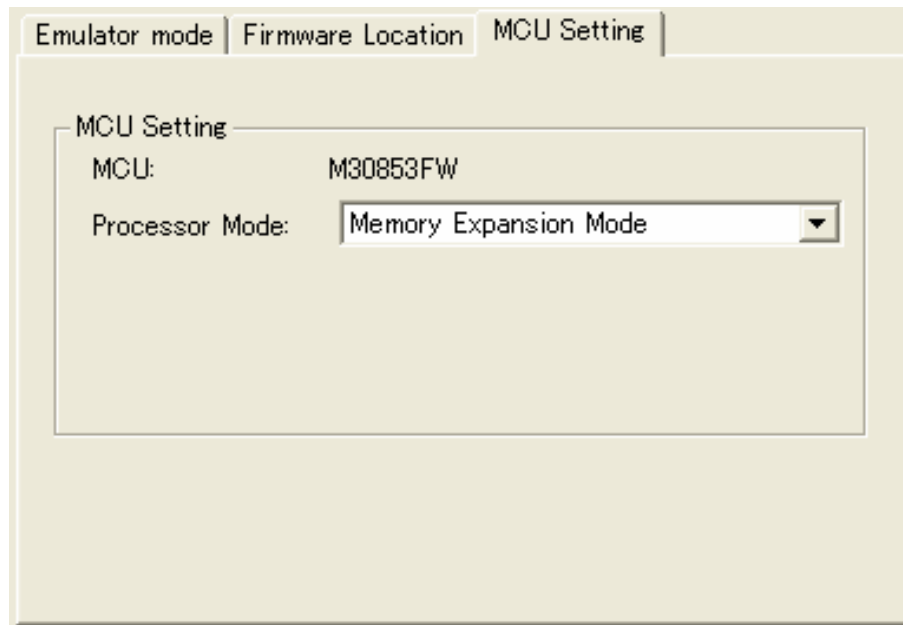


Figure 6.3 [MCU Setting] Tab

Specify the processor mode

Specify the processor mode according to the user system. One of the followings can be specified.

- Single-Chip Mode
- Memory Expansion Mode

Notes:

The following describes the precautions to be taken when using the emulator in memory expansion mode.

- * When the external area cannot be rewritten via normal memory access, software breaks cannot be used in that area.

Section 7 Applicable Tool Chain and Partner Tools

With the M32C/80 Series E8 emulator, you can debug a module created by the inhouse tool chain and third-party products listed in Table 7.1 below.

Table 7.1 Applicable Tool Chain and Partner Tools

Tool chain	M3T-NC308WA V.5.20 Release 02 or later
Partner tools	IAR EWM32C V.3.10A or later

E8 Emulator

Additional Document for User's Manual

Notes on Connecting the M32C/84, M32C/85, M32C/86, M32C/87 and M32C/88

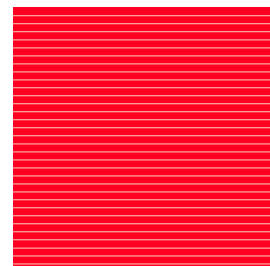
Publication Date: August 01, 2005 Rev.1.00
 November 01, 2006 Rev.4.00

Published by: Sales Strategic Planning Div.
 Renesas Technology Corp.

Edited by: Microcomputer Tool Development Department
 Renesas Solutions Corp.

© 2006. Renesas Technology Corp. and Renesas Solutions Corp., All rights reserved. Printed in Japan.

E8 Emulator Additional Document for User's Manual



Renesas Technology Corp.
2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan